



# Papyrus for Information Modeling

Experience report on creating a modeling tool based on Papyrus

Philip Langer

[planger@eclipsesource.com](mailto:planger@eclipsesource.com)

Maximilian Koegel

[mkoegel@eclipsesource.com](mailto:mkoegel@eclipsesource.com)

## Context: Domain-specific Modeling Tools

- Domain-specific modeling
  - Highly specialized domain-specific modeling language (DSML)
  - Specific needs of a domain, its users, and their workflows
- DSML & modeling tool
  - One product
  - Streamlined editors
  - Including domain-specific functionality for
    - Simulation
    - Code generation
    - Verification and Validation, ...
  - But exclude what is not needed
  - Efficiency of the domain user

*There won't be “the one DSML tool”, but “the one DSML tool **platform**”*

## Context: Domain-specific Modeling Tools with Papyrus

- Papyrus is a modeling **platform**
  - Standard-conform UML support & DSMLs with UML Profiles
  - Customizable diagram, table, and tree views (concrete syntax)
  - Customizable tooling & modeling environment
- Papyrus is built on the Eclipse platform
  - Mature and powerful platform
    - Rich-client products
    - Engineering tools
    - Integrated dev. environments
  - Easy to customize
  - Easy to extend
  - Huge ecosystem



# Context: Domain-specific Modeling Tools with Papyrus

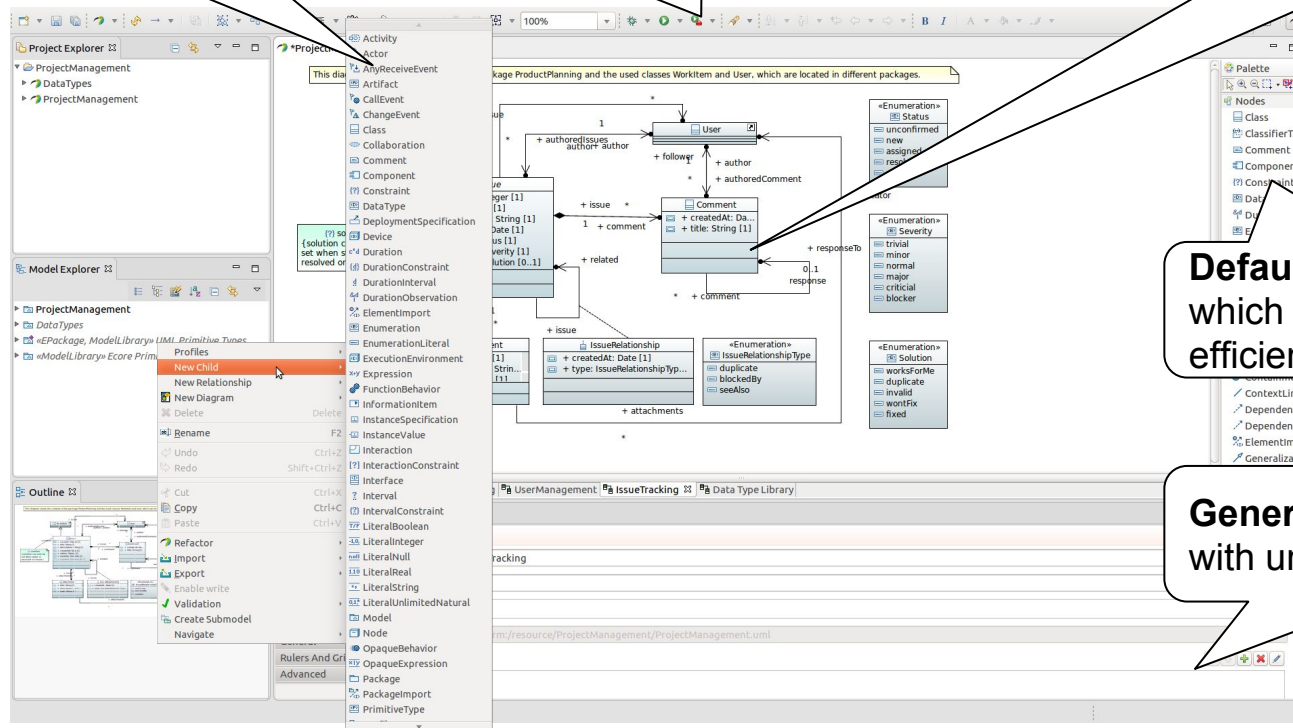
**Huge menus with elements which are not part of your domain**

**Bloated toolbars with unneeded functionality**

**Suboptimal diagram styles which are not optimized for your use case**

**Default palette tools which don't facilitate efficiency for your domain**

**Generic property sheets with unnecessary features**



# Context: Domain-specific Modeling Tools with Papyrus

**Huge menus** with elements which are not part of your domain

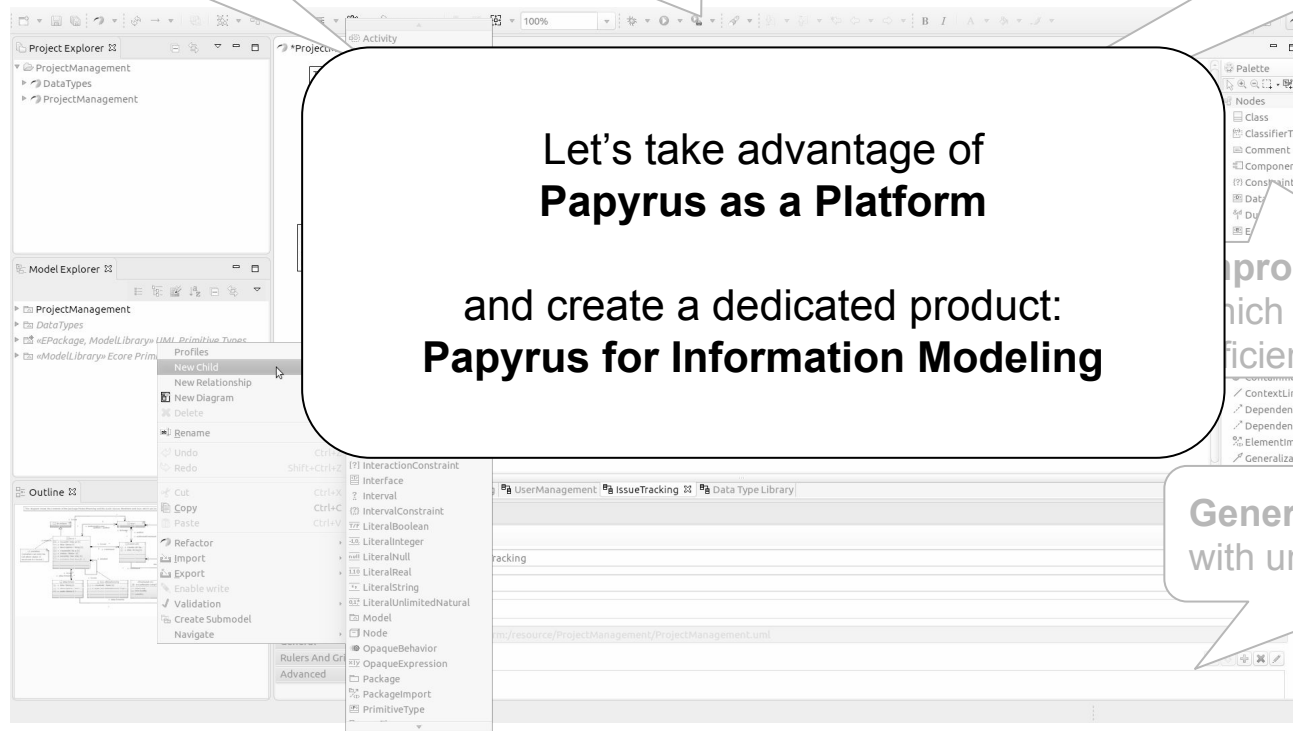
**Bloated toolbars** with unneeded functionality

**Suboptimal diagram styles** which are not optimized for your use case

Let's take advantage of  
**Papyrus as a Platform**  
and create a dedicated product:  
**Papyrus for Information Modeling**

**Improper palette tools** which don't facilitate efficiency for your domain

**Generic property sheets** with unnecessary features



## Context: Papyrus for Information Modeling

- Objective: Information Modeling product
  - Based on the Papyrus platform
  - Streamlined for information modeling with UML class diagrams
  - Optimize ease of use for users new to Papyrus
  - Reduce the complexity of Papyrus and the Eclipse UI
  - Illustrate the customizability of Papyrus and the Eclipse platform
    - Blueprint for future Papyrus-based products
- Project details
  - Carried out by CEA and EclipseSource
  - Project size was 36 developer days
  - Project started mid May
  - Version 0.7.1 released beginning of July
- Funded by the Papyrus Industrial Consortium
  - <https://www.polarsys.org/ic/papyrus>

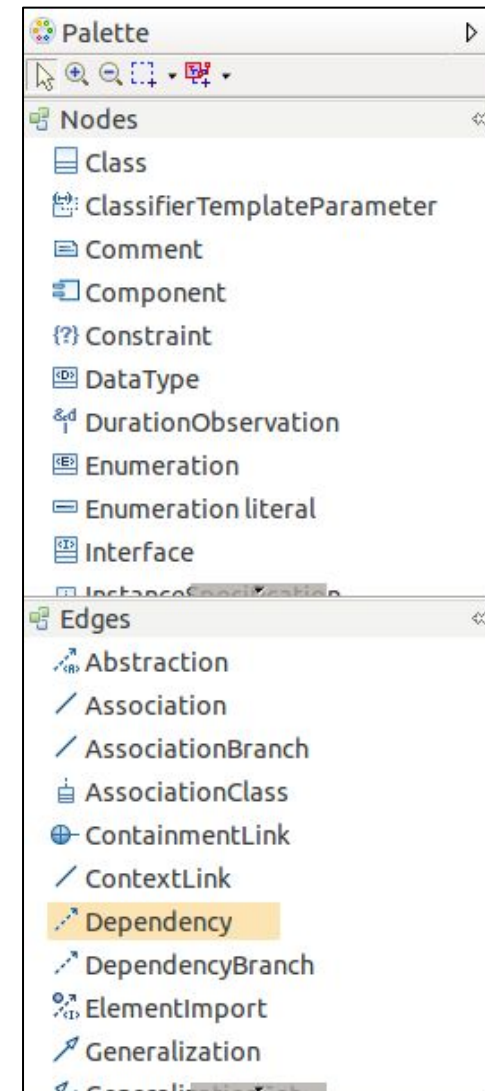
## Work done so far

- Identification of the UML subset used for Information Modeling
  - Based on sample models we retrieved
- Implementation of the UML subset as a Papyrus viewpoint
  - Streamline the Papyrus user interfaces
  - Palette, property views, etc.
- Implementation of the simplification of the Eclipse UI
  - Toolbars, wizards, menu actions, etc.
- Implementation of a diagram style
  - Optimized for information modeling
- Documentation
  - Importable example model
  - Getting started tutorial based on Eclipse Cheat Sheets
  - Documentation of how to create a Papyrus-based product
- Technical tasks excluded in the list above
  - Setup of continuous integration build and development infrastructure, product configuration and target platform specification, separation of composable and non-composable plug-ins, code reviews for best practices, ...

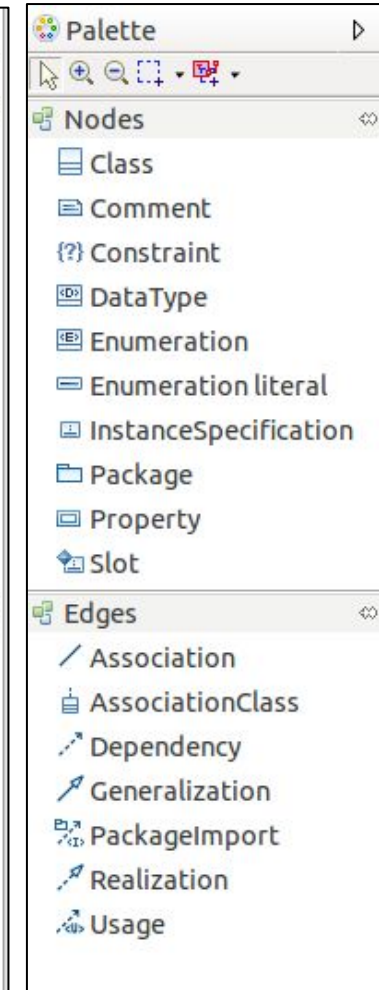
## Work done so far

- Simplification: Papyrus UI
  - Palette

Before



After

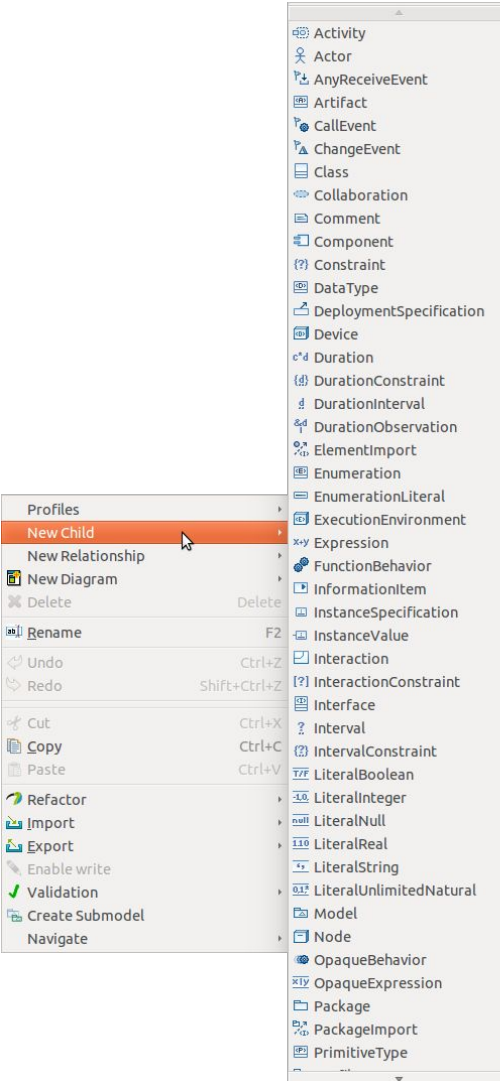




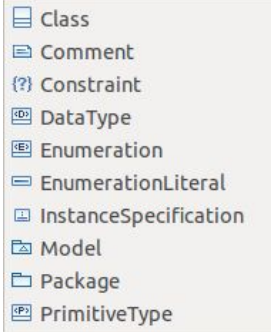
## Work done so far

- Simplification: Papyrus UI
  - New child creation menus

**Before**

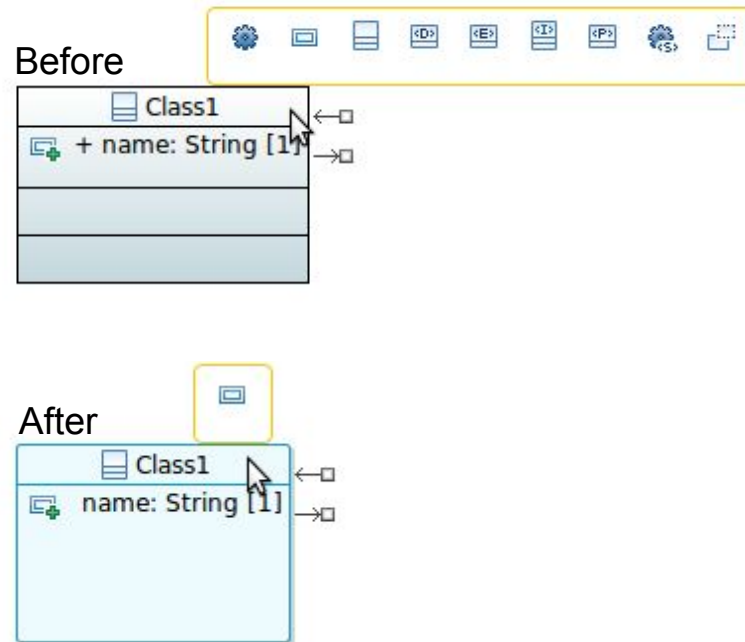


**After**



## Work done so far

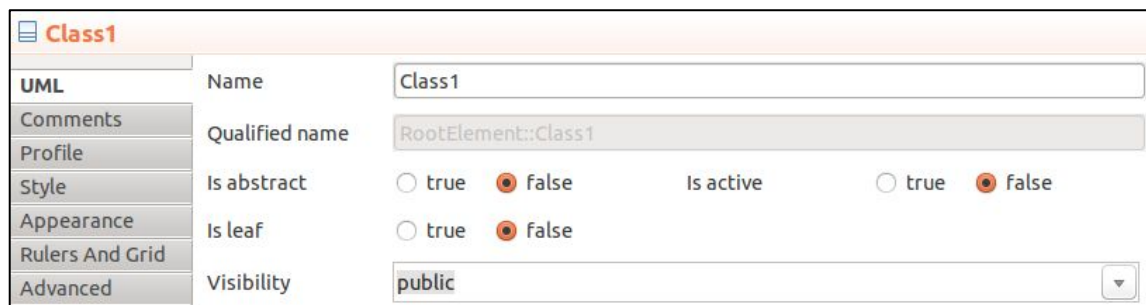
- Simplification: Papyrus UI
  - Modeling assistants and diagram style



## Work done so far

- Simplification: Papyrus UI
  - Property views

Before



Class1	
<b>UML</b>	Name: Class1
Comments	Qualified name: RootElement::Class1
Profile	Is abstract: <input type="radio"/> true <input checked="" type="radio"/> false      Is active: <input type="radio"/> true <input checked="" type="radio"/> false
Style	Is leaf: <input type="radio"/> true <input checked="" type="radio"/> false
Appearance	Visibility: public
Rulers And Grid	
Advanced	

After

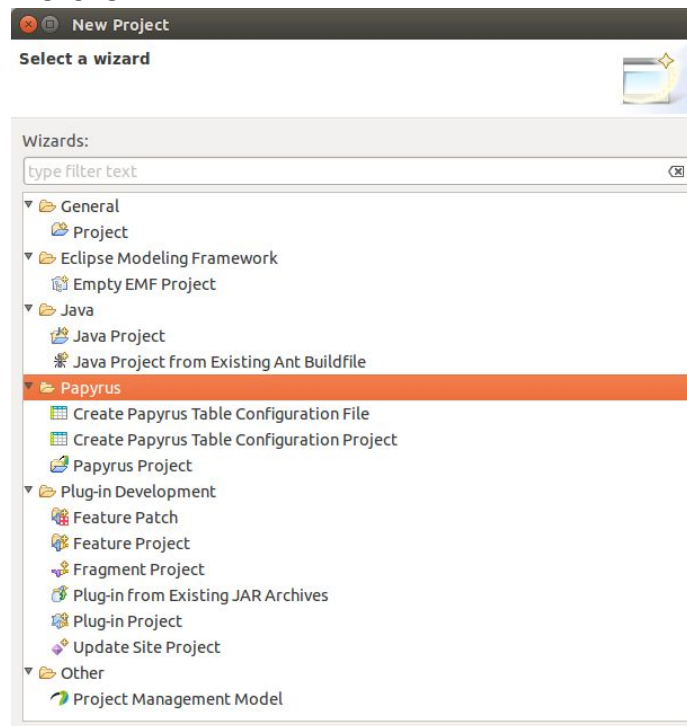


Class1	
<b>UML</b>	Name: Class1
Comments	Qualified name: RootElement::Class1
Profile	Is abstract: <input type="radio"/> true <input checked="" type="radio"/> false      Visibility: public
Style	
Appearance	Owned attribute: name : String
Rulers And Grid	
Advanced	

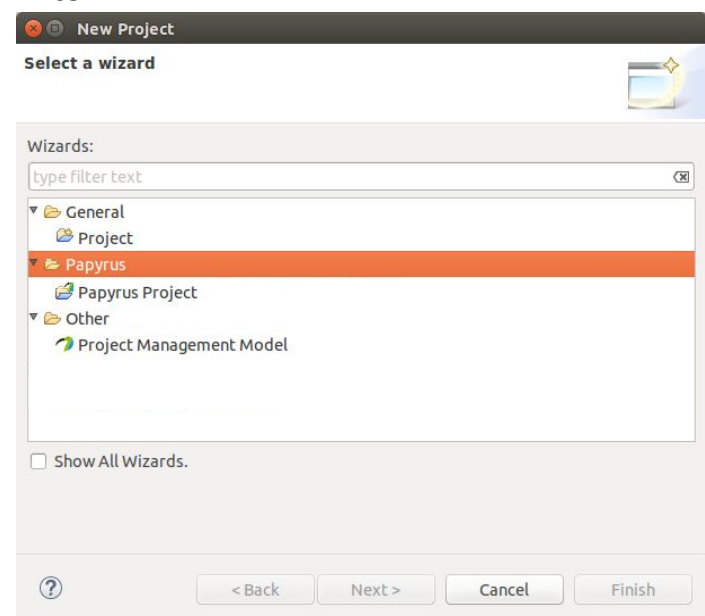
## Work done so far

- Simplification: Eclipse UI
  - Wizards

Before



After

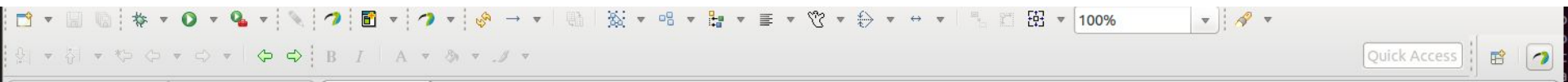


## Work done so far

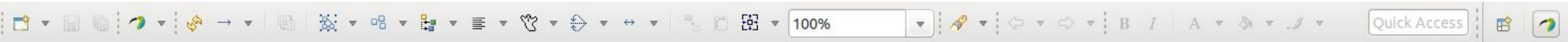
- Simplification: Eclipse UI

- Toolbars

Before

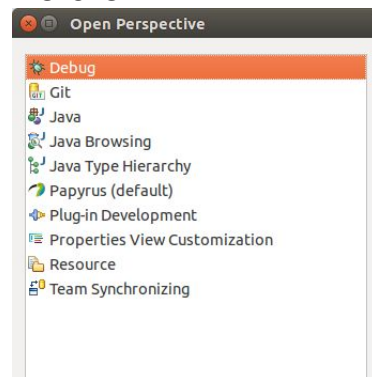


After

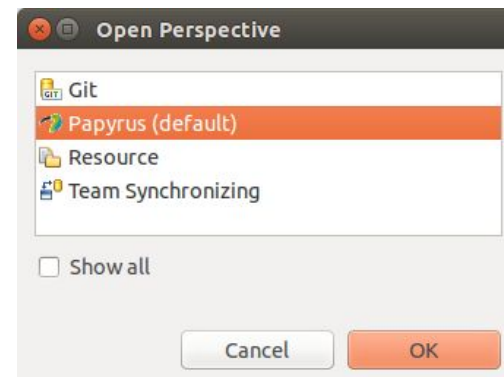


- Perspectives

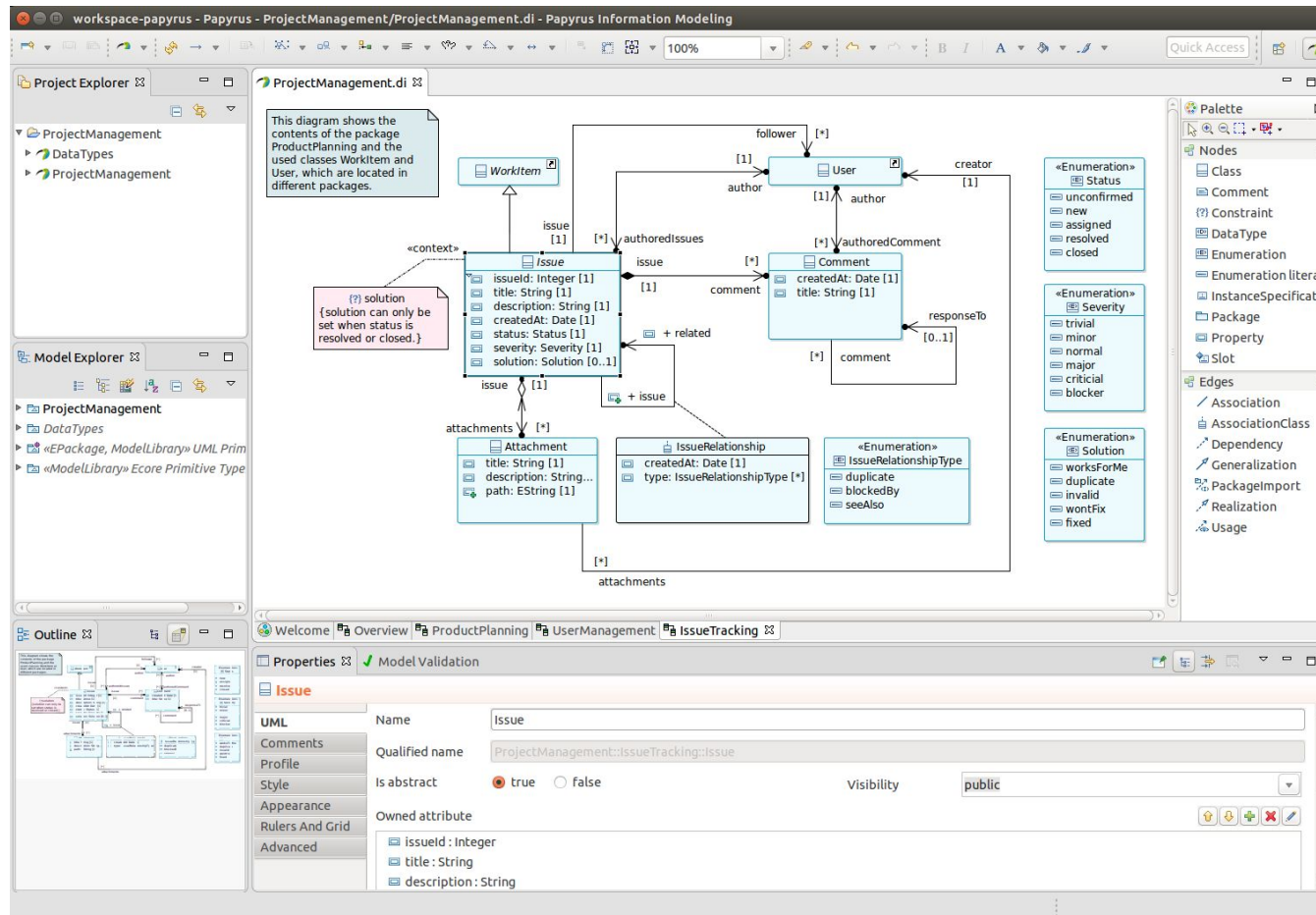
Before



After



# Demo

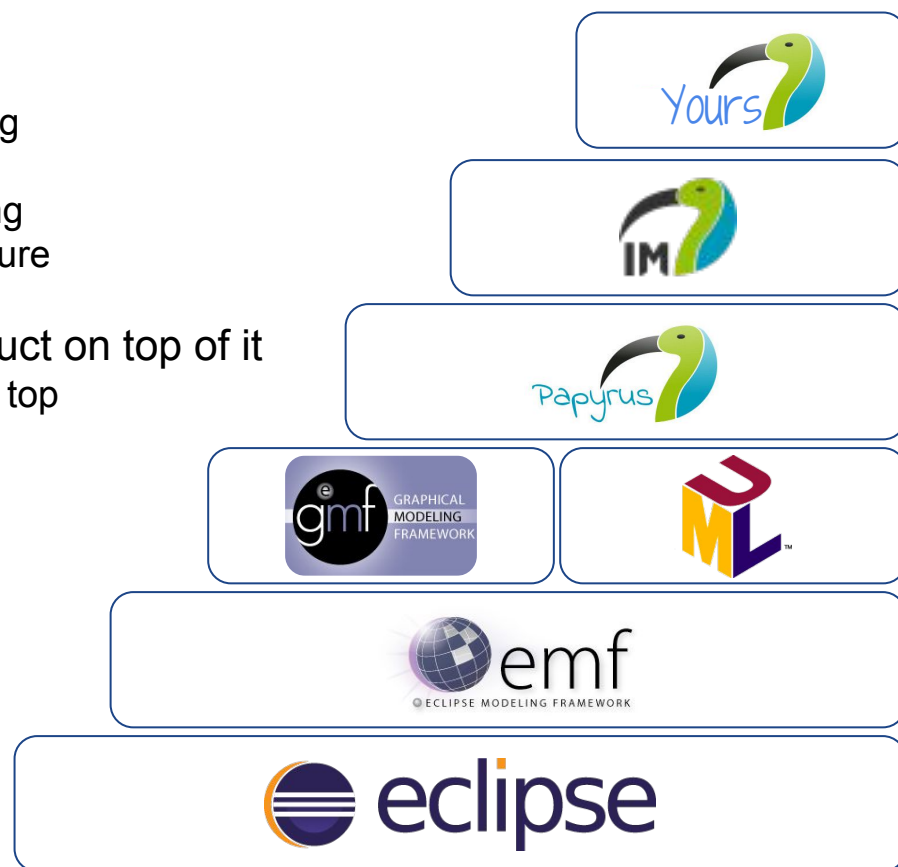


## Next Steps and Future Directions

- Get your feedback on requirements and enhancements
  - Create a backlog for future development
    - Dedicated UML profiles?
    - Specific code generators?
    - GenDoc integration?
  - Find partners to support us in the future development
  - All contributions are more than welcome
    - Requirements
    - Testing
    - Funding, etc.
- Ease definition of Papyrus-based products
  - Several different configuration models involved for customization
  - Combine this information including the Eclipse UI customization:  
One Papyrus-product configuration
  - Archetype for a Papyrus-based product build

## Next Steps and Future Directions

- Recursive customization
  - Take a base product
    - Papyrus for Information Modeling
    - Papyrus for Real-Time
    - Papyrus for Systems Engineering
    - Papyrus for Enterprise Architecture
  - Build your custom modeling product on top of it
    - Define your language subset on top
    - Add your own domain concepts
    - Add your additional functionality





# Invitation to Contribute

License EPL 1.0

- Get Papyrus for Information Modeling
  - Release 0.7.1 available (see [wikipage](#))
- Feedback, requirements, and bugs
  - Contact us ([planger@eclipsesource.com](mailto:planger@eclipsesource.com))
  - Raise bug reports ([bugzilla](#))
  - Papyrus mailing list ([mdt-papyrus.dev](mailto:mdt-papyrus.dev))
- Fixes and enhancements
  - Push for review via [gerrit](#)
  - See also contributor guidelines (view [wikipage](#))
- We offer professional services
  - Support
  - Custom enhancements
  - Sponsored OS development
  - Come to our booth (Zeligsoft & EclipseSource)

## Related Sessions

Tomorrow 10:30

Papyrus-RT out of its shell  
Charles Rivet (Zeligsoft)

Tomorrow 11:30

Customizable collaborative  
modeling for DSMLs  
M. Koegel & P. Langer

Tomorrow 13:45

Papyrus Customization and  
DSML support  
Benoit Maggi (CEA)